

LA-UR-21-25835

Approved for public release; distribution is unlimited.

Title: Threshold-Based Quantum Optimization

Author(s): Golden, John Kimbell
Baertschi, Andreas
O'Malley, Daniel
Eidenbenz, Stephan Johannes

Intended for: IEEE International Conference on Quantum Computing and Engineering,
2021-10-18/2021-10-22 (Virtual, New Mexico, United States)
arXiv

Issued: 2021-11-10 (rev.1)

Disclaimer:

Los Alamos National Laboratory, an affirmative action/equal opportunity employer, is operated by Triad National Security, LLC for the National Nuclear Security Administration of U.S. Department of Energy under contract 89233218CNA000001. By approving this article, the publisher recognizes that the U.S. Government retains nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U.S. Government purposes. Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy. Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness.

Threshold-Based Quantum Optimization

John Golden*, Andreas Bärttschi*, Daniel O'Malley† and Stephan Eidenbenz*

* *CCS-3 Information Sciences, Los Alamos National Laboratory, Los Alamos, NM 87544, USA*

Email: golden@lanl.gov, baertschi@lanl.gov, eidenben@lanl.gov

† *EES-16 Earth Sciences, Los Alamos National Laboratory, Los Alamos, NM 87544, USA*

Email: omalled@lanl.gov

Abstract—We propose and study Th-QAOA (pronounced Threshold QAOA), a variation of the Quantum Alternating Operator Ansatz (QAOA) that replaces the standard phase separator operator, which encodes the objective function, with a threshold function that returns a value 1 for solutions with an objective value above the threshold and a 0 otherwise. We vary the threshold value to arrive at a quantum optimization algorithm. We focus on a combination with the Grover Mixer operator; the resulting GM-Th-QAOA can be viewed as a generalization of Grover's quantum search algorithm and its minimum/maximum finding cousin to *approximate* optimization.

Our main findings include: (i) we provide intuitive arguments and show empirically that the optimum parameter values of GM-Th-QAOA (angles and threshold value) can be found with $O(\log(p) \times \log M)$ iterations of the classical outer loop, where p is the number of QAOA rounds and M is an upper bound on the solution value (often the number of vertices or edges in an input graph), thus eliminating the notorious outer-loop parameter finding issue of other QAOA algorithms; (ii) GM-Th-QAOA can be simulated classically with little effort up to 100 qubits through a set of tricks that cut down memory requirements; (iii) somewhat surprisingly, GM-Th-QAOA outperforms non-thresholded GM-QAOA in terms of approximation ratios achieved. This third result holds across a range of optimization problems (MaxCut, Max k-VertexCover, Max k-DensestSubgraph, MaxBisection) and various experimental design parameters, such as different input edge densities and constraint sizes.

I. INTRODUCTION

Using the power of quantum computing to solve combinatorial optimization problems, such as Minimum Traveling Salesperson, Maximum Satisfiability, or Maximum Cut, has been one of the main drivers of the development of quantum computing theory and practice. Early hopes (mid-1990s) of using quantum computers to solve NP-hard optimization problems in polynomial time had to be tempered after the discovery of oracles relative to which NP is not contained in BQP [3]; thus, the existence of polynomial-time quantum algorithms for NP-complete problems is highly unlikely albeit not impossible. Polynomial factor speed-ups over the best known classical alternatives are an active area of quantum algorithms research that – often as a side effect – expose new facets of the boundary between NP and BQP .

Defined for general gate-level quantum computing, the Quantum Approximate Optimization Algorithm (QAOA), [9],

which was later generalized to the Quantum Alternating Operator Ansatz (QAOA) [14], can be seen as low-order Trotterization of adiabatic computing with promising initial results in the form of provable approximation guarantees for MaxCut on 3-regular graphs [9] and the E3Lin2 problem on bounded occurrence instances [10]. Additional provable QAOA-based approximation ratio guarantees have been largely elusive until now, but the value of QAOA as an optimization heuristic is immense. Akin to classical optimization heuristics, properties other than approximation ratio guarantees need to be studied: circuit depth, convergence guarantees, average-case performance on random or constructed instances, outer loop parameter finding methods, etc.

The seminal work of the Quantum Alternating Operator Ansatz (QAOA) [14] defines a framework that allows modeling of almost any combinatorial optimization problem as a QAOA problem with the objective functions as the sum of Pauli operator-based Hamiltonians forming the phase separator operator and a few variations of mixing unitaries. Follow-on work has focused on ways of efficiently finding outer loop parameters, e.g., [16], the efficient preparation of high-quality initial states such as Dicke states [2], Grover-inspired mixer unitaries [1], [5], experimental studies of more optimization problems [7], and studies on the XY-mixer model [19].

In this paper, we propose Th-QAOA (pronounced Threshold QAOA) – a variation of the Quantum Alternating Operator Ansatz (QAOA) that replaces the standard phase separator operator, which encodes the objective function, with a threshold function that returns a value 1 for solutions with an objective value above the threshold and a 0 otherwise (Section II). We vary the threshold value to arrive at a quantum approximate optimization algorithm. Th-QAOA can be combined with any of the previously studied Mixers for QAOA (e.g., Transverse Field based X -mixer [9], [10] for unconstrained problems, XY -model Ring and Clique Mixers for Hamming weight-constrained problems [7], [19]); in this paper we focus on the combination with Grover Mixers (which have been introduced for both unconstrained [1] and constrained [5] optimization problems), which we denote by GM-Th-QAOA.

We first show in Section III through a mix of formal analysis and experimental evidence that the optimum values for of GM-Th-QAOA, namely the threshold value th as well as the angles β_i, γ_i for each QAOA round $i \leq p$ with p being the total number of QAOA rounds, can be found with a simple algorithm in $O(\log p \times \log M)$ iterations of the classical outer

loop, where M is an upper bound on the solution value. In the graph or logical optimization problems that we consider, M is usually of the same order as the input size, e.g., the number of vertices or edges in an input graph for problems such as MaximumCut. Our parameter finding algorithm thus eliminates the notorious outer-loop parameter finding issue of other QAOA algorithms. Our algorithm relies on insights from numerical simulation experiments that show that as we increase the threshold value (under fixed round count), the best approximation ratio that GM-Th-QAOA finds first increases monotonically, reaching a peak and then decreasing. This stable shape allows us to find the peak value with the optimum threshold value via an exponential search.

As a second contribution, we describe (Section IV) a method of efficiently simulating GM-Th-QAOA circuits classically (after an initial precomputation of objective values) whenever we are only interested in calculating the expectation value of the objective function of the selected solution, which is all that we need to assess the performance of GM-Th-QAOA. Our method works particularly well when coupled with the Grover Mixer [5] and used on constrained optimization problems. Constrained optimization problems are problems where not all computational basis states correspond to feasible solutions, but the set of feasible solutions is constrained in some ways, such as only including solutions with exactly k qubits set to 1. Examples of constrained problems are Max k -Densest Subgraph, which asks for a subset V' of exactly k of the vertices of an input graph such that a maximum number of edges have both end vertices in set V' , k -VertexCover, Max Bisection (which we define later); MaximumCut is an example of an unconstrained problem. Our simulation method limits the state vector calculation to the subspace of feasible solutions. In addition, leveraging a feature of the Grover Mixer that solutions of equal objective function value all are assigned the same amplitude, we can further reduce our search space to all possible objective function values, which is usually M as we are dealing with discrete values (such as counting edges). Our simulation methods allow us to simulate high round counts nearly effortlessly (compared to the precomputation) and to go to large problem sizes (90+ vertices) with relative ease, giving us much greater confidence that we may indeed be observing performance trends that can be extrapolated to larger problem instances, particularly when compared to the standard problem size in classical QAOA simulations (in most other works, including our own), which is around graph size 10–16.

In our third contribution (Section V), we present a large study of experimental results comparing GM-Th-QAOA with a Grover Mixer QAOA using the standard objective-value based phase separator. Our experiments are for Maximum k -Densest Subgraph, Maximum k -Vertex Cover, Maximum Bisection, and Maximum Cut, with graph sizes ranging up to 90, three different edge probabilities (in a random graph model), and three different constraint values (parameter k) where applicable. Our main experimental finding initially surprised us: Threshold QAOA with the Grover Mixer outperforms the traditional objective-value based QAOA with Grover Mixer across

all optimization problems and across all parameter values that we tested. This experimental result, coupled with the fact that we can find optimum outer loop parameter values much faster than for standard QAOA, makes GM-Th-QAOA a strong contender for best-in-class quantum optimization heuristic. In addition, our experiments validate the scalability of our fast simulation method. Finally, we observe that increasing the number of rounds beyond polynomial in input size leads GM-Th-QAOA to degenerate into Grover search, which – on the positive side – guarantees Grover-style speed-up, and – on the negative side – makes it questionable that QAOA’s worst-case running time guarantee can be better than Grover’s, or in other words, QAOA maybe no more effective than Grover’s unstructured search in the worst case. While we appear to see such exponential blow-up of round counts in a few examples, most real-life examples are far from worst-case, however. Thus we conclude that GM-Th-QAOA is a very promising quantum heuristic for finding high-quality approximate solutions for optimization problems.

II. THRESHOLD QAOA

A. Definition

Suppose we are given an instance I of a combinatorial optimization problem over inputs $x \in S$, where $S \subseteq \{0, 1\}^n$ is the set of all feasible solutions; let $C(x): S \rightarrow \mathbb{R}$ be the objective function which evaluates the cost of solution x . W.l.o.g., we look at maximization problems (as opposed to minimization), where we want to find maximum or close-to-maximum value solutions. I could be an instance of Maximum Satisfiability, defined on n binary variables with, say m clauses. If all variable allocations are feasible solutions (i.e., $S = \{0, 1\}^n$), we say that the problem is unconstrained, such as Maximum Satisfiability; if $S \subsetneq \{0, 1\}^n$, we call the problem constrained, such as Max k -Densest Subgraph, which asks for a set of k vertices from a given graph with a maximum number of edges with edges in the induced subgraph. Let

$$\delta(x) = \begin{cases} 0 & \text{if } C(x) \leq th \\ 1 & \text{otherwise} \end{cases} \quad (1)$$

be an indicator function that assigns a value of 1 to all solutions with an objective function above threshold th and zero otherwise. The quantum subroutine of Th-QAOA is defined on an input tuple $(I, th, |\psi\rangle, H_P, H_M, p, \vec{\beta}, \vec{\gamma})$, where:

- $|\psi\rangle$ is the initial state,
- $H_P |x\rangle = \delta(x) |x\rangle$ is a phase separator Hamiltonian,
- H_M is a mixer Hamiltonian,
- p is the number of rounds/levels to run the algorithm, and
- $\vec{\gamma} = (\gamma_1, \dots, \gamma_p)^T$ and $\vec{\beta} = (\beta_1, \dots, \beta_p)^T$, each of length p two real vectors. These values are often called angles.

The Th-QAOA algorithm returns a quantum state after preparing the initial state $|\psi\rangle$, and applying p rounds of the alternat-

ing simulation of the phase separator Hamiltonian for time γ_i and the mixer Hamiltonian for time β_i :

$$|\text{Th-QAOA}\rangle = \underbrace{e^{-i\beta_p H_M} e^{-i\gamma_p H_P}}_{\text{round } p} \dots \underbrace{e^{-i\beta_1 H_M} e^{-i\gamma_1 H_P}}_{\text{round 1}} |\psi\rangle \quad (2)$$

In each round, H_P is applied first, which separates the basis states of the state vector by phases $e^{-i\gamma\delta(x)}$. The mixing operator H_M then provides parameterized interference between solutions of different cost values. After p rounds, the state $|\text{Th-QAOA}\rangle$ is measured in the computational basis and returns a sample solution y of cost value $C(y)$ with probability $|\langle y|\text{Th-QAOA}\rangle|^2$. The only difference to standard QAOA is our definition of the phase separator H_P through a threshold delta function $\delta(x)$ instead of the cost value $C(x)$. The threshold parameter th should only take on values below a known upper bound of the combinatorial problem, e.g., $th \leq m$ in our MaxSat problem; for most problems (with the notable exception of number problems, such as Min Traveling Salesperson), th can be bounded polynomially in the size of the input problem, which will allow us to search for optimum threshold values in polynomial or even logarithmic outer loops.

Although we use $\delta(x)$ as the function in the phase separator, we still evaluate the quality of the expectation value over the original objective function C as $\langle \text{Th-QAOA} | C | \text{Th-QAOA} \rangle$. For a maximization problem, we say Th-QAOA achieves an approximation ratio $\frac{\langle \text{Th-QAOA} | C | \text{Th-QAOA} \rangle}{\max_{x \in S} C(x)}$.

B. Connection to Grover's algorithm

The QAOA framework, including Th-QAOA, can be applied to a wide variety of problems and problem-specific mixers [14]. Here we use the Grover Mixer, which has been studied for both unconstrained [1] and constrained problems [5], and can be used whenever there exists an efficient state preparation unitary U_S that prepares the equal superposition of all feasible states S , $|S\rangle := |S|^{-1/2} \sum_{x \in S} |x\rangle$. Then we have a Grover Mixer Hamiltonian $H_{GM} := |S\rangle\langle S|$ with Grover Mixer [5]

$$\begin{aligned} e^{-i\gamma H_{GM}} &= Id - (1 - e^{-i\beta}) |S\rangle\langle S| \\ &= U_S (Id - (1 - e^{-i\beta}) |0\rangle\langle 0|) U_S^\dagger. \end{aligned} \quad (3)$$

We call this combination of a threshold-based phase separator with the Grover Mixer GM-Th-QAOA, and obtain an algorithm that shares many traits with Grover's quantum search algorithm [12] and both variational versions thereof [13], [17], [20] and its minimum/maximum finding version [8]. We thus bring these purely search-oriented algorithms into the approximate optimization realm, similar to the original Quantum Approximate Optimization Algorithm which was inspired by the minimum/maximum finding adiabatic algorithm. See the Related Works section VI for more details on this connection.

C. Completing the Picture: Outer Loops

For any practical use, the core quantum subroutine of the Threshold QAOA algorithm as described needs to be embedded in a classical outer loop algorithms to find good values for input parameters th , $\vec{\beta}$, and $\vec{\gamma}$. Quickly finding good

values for these variational parameters has been a challenge and the focus of a sizable fraction of the QAOA literature, see e.g. [9], [10], [18], [19], [22].

For a reverse picture of the GM-Th-QAOA with an arbitrary but fixed threshold th and a *variational* number of rounds p values, we provide intuition through an analysis of solution distributions and backed up by experiments that the optimum angle parameters $\vec{\beta}$, and $\vec{\gamma}$ can be set to $\beta_i = \gamma_i = \pi$ for all rounds $i < p$, and the values for β_p and γ_p can be found through a linear search.

For a fixed number of rounds p as is common in the QAOA framework, we can find the optimum threshold value th^* for GM-Th-QAOA, which we define as the value that maximizes the achieved approximation ratio r , through a modified binary search because the approximation ratios monotonically increase before monotonically decreasing again with increasing threshold value resulting in a single peak (see Figure 1 for an example). Combining this with the observation above, we find that for small p the optimal parameters will be $\beta_i = \gamma_i$ for all rounds i , while for larger p we can again deploy binary search for a transition round t such that $\beta_i = \gamma_i = \pi \forall i < t$ and $\beta_i = \gamma_i = 0 \forall i > t$, with angles β_t, γ_t that can be found with a fine grid search.

The ease of finding these values in an outer loop is a key advantage of GM-Th-QAOA over other variations. The total running time of GM-Th-QAOA is thus the product of

- the quantum subroutine running time $O(p(t_{GM} + t_{PS}))$, where $t_{GM}, t_{PS} \in \text{poly}(M)$ are the circuit depths of the Grover mixer and phase separator unitaries,
- $\text{poly}(M)$ iterations thereof (for fixed parameters th, β, γ) to reliably estimate $\langle \text{GM-Th-QAOA} | C | \text{GM-Th-QAOA} \rangle$, where the exact polynomial depends on the concentration of sample distribution from $|\text{GM-Th-QAOA}\rangle$ [7], [9],
- $O(\log p)$ time to find the transition round t and angles β_t, γ_t for a candidate threshold th ,
- $O(\log M)$ time to find the optimum threshold value th^* .

We show in Section IV that after an initial precomputation of the objective values of all feasible states S , the first two points can be classically simulated efficiently.

III. FINDING HIGH-QUALITY PARAMETER VALUES EFFICIENTLY FOR GM-TH-QAOA

We use a structural characterization of the distribution of objective values among feasible solutions to build a simple heuristic for finding high-quality angle values for GM-Th-QAOA. For ease of presentation, we adopt the convention that the GM-Th-QAOA phase separator only acts on states strictly greater the threshold, rather than greater than or equal to the threshold. Therefore, if our optimization instance I has a best solution x^* with objective value $C(x^*)$, the threshold that will select states of this score is $C(x^*) - 1$. which we will call the **maximal threshold**. Moreover, as the number of rounds decreases, the threshold which returns the highest approximation ratio also decreases. Therefore, the maximal threshold is often not the threshold that returns the highest approximation ratio. We thus call the threshold value th^* ,

which returns the highest approximation ratio for a given number of rounds p the **optimal threshold**.

We begin with some notation: after p rounds of Th-QAOA with threshold th , we have the state

$$|S^{(p)}\rangle = c_0^{(p)} \left(\sum \text{states with score} \leq th \right) + c_1^{(p)} \left(\sum \text{states with score} > th \right) \quad (4)$$

with coefficients at round 0 as $c_0^{(0)} = c_1^{(0)} := |S|^{-1/2}$ for $|S|$ feasible solutions. For a larger number of rounds p , we have the recursive form:

$$\begin{aligned} c_0^{(p)} &= c_0^{(p-1)} - (1 - e^{-i\beta}) \left(r c_0^{(p-1)} + (1-r) c_1^{(p-1)} e^{-i\gamma} \right), \\ c_1^{(p)} &= c_1^{(p-1)} e^{-i\gamma} - (1 - e^{-i\beta}) \left(r c_0^{(p-1)} + (1-r) c_1^{(p-1)} e^{-i\gamma} \right) \end{aligned} \quad (5)$$

Here, we have used the basic definitions of the Th-QAOA algorithm (2) and the Grover Mixer (3) and dropped indices $p-1$ for the angle labels to ease notation. Label r denotes the fraction of states with score $\leq th$. These states have degeneracy d_0 and d_1 , respectively, and we have $d_0 + d_1 = |S|$. Thus, using $r \equiv \frac{d_0}{|S|}$, we arrive at Eq. (5).

A. First-round Angle Values

Let us start by exploring the first round in detail. We have

$$\begin{aligned} c_0^{(1)} &= |S|^{-1/2} (1 - (1 - e^{-i\beta}) (r + (1-r) e^{-i\gamma})), \\ c_1^{(1)} &= |S|^{-1/2} (e^{-i\gamma} - (1 - e^{-i\beta}) (r + (1-r) e^{-i\gamma})), \end{aligned} \quad (6)$$

Taking the absolute value squared of these coefficients gives

$$\|c_0^{(1)}\|^2 = |S| (1 + 2(r-1)(\sin(\beta)\sin(\gamma) + 4(2r-1)\sin^2(\beta/2)\sin^2(\gamma/2))), \quad (7)$$

$$\|c_1^{(1)}\|^2 = |S| (1 + 2r(\sin(\beta)\sin(\gamma) + 4(2r-1)\sin^2(\beta/2)\sin^2(\gamma/2))) \quad (8)$$

Observation 1. When $r < \frac{3}{4}$,

$$\beta = \gamma = \arctan(-\sqrt{3-4r}, 1-2r) \quad (9)$$

gives $\|c_0^{(1)}\|^2 = 0$, where $\arctan(x, y)$ takes into account which quadrant the point (y, x) is in when calculating $\arctan(x/y)$.

Observation 2. For $r \geq \frac{3}{4}$, $\beta = \gamma = \pi$ minimizes $\|c_0^{(1)}\|^2$ and maximizes $\|c_1^{(1)}\|^2$ and $\langle \psi^{(1)} | H_P | \psi^{(1)} \rangle$.

In other words, when $r < \frac{3}{4}$ there exist β, γ that will entirely kill off all states with score $\leq th$, and no further rounds are necessary. If $r \geq \frac{3}{4}$, setting $\beta = \gamma = \pi$ gives the highest approximation ratio possible after round 1, and further rounds are necessary to achieve $\|c_0\|^2 = 0$.

Observation 1 can be verified by explicit calculation. For Observation 2, we give the following proof.

Proof. Our goal is to maximize

$$\|c_1^{(1)}\|^2 = |S| (1 + 2r(\sin(\beta)\sin(\gamma) + 4(2r-1)\sin^2(\beta/2)\sin^2(\gamma/2))) \quad (10)$$

for $r \geq 3/4$. We re-write $r = 3/4 + \epsilon/8$ for $\epsilon \geq 0$, and drop constant terms and overall factors to get the function we are trying to maximize:

$$\sin(\beta)\sin(\gamma) + (2 + \epsilon)\sin^2(\beta/2)\sin^2(\gamma/2). \quad (11)$$

Our claim is that $\beta = \gamma = \pi$ are optimum values, i.e. our goal is now to prove

$$\sin(\beta)\sin(\gamma) + (2 + \epsilon)\sin^2(\beta/2)\sin^2(\gamma/2) \leq 2 + \epsilon, \quad (12)$$

or, re-arranging terms,

$$\begin{aligned} \sin^2(\beta/2)\sin^2(\gamma/2) + \frac{1}{2}\sin(\beta)\sin(\gamma) \leq \\ 1 + \frac{\epsilon}{2}(1 - \sin^2(\beta/2)\sin^2(\gamma/2)). \end{aligned} \quad (13)$$

Consecutively using on the left side the trigonometric identities $\sin(2\theta) = 2\sin(\theta)\cos(\theta)$ and $\cos(\alpha \pm \beta) = \cos(\alpha)\cos(\beta) \mp \sin(\alpha)\sin(\beta)$, we bring Eq. (13) into the equivalent form

$$\begin{aligned} \cos(\beta/2 - \gamma/2)^2 - \cos^2(\beta/2)\cos^2(\gamma/2) \leq \\ 1 + \frac{\epsilon}{2}(1 - \sin^2(\beta/2)\sin^2(\gamma/2)), \end{aligned} \quad (14)$$

which holds as $0 \leq \frac{\epsilon}{2}(1 - \sin^2(\beta/2)\sin^2(\gamma/2))$ as well as $0 \leq \cos(\beta/2 - \gamma/2)^2 \leq 1$; $0 \leq \cos^2(\beta/2)\cos^2(\gamma/2) \leq 1$. \square

B. Extending Beyond the First Round

Based on numerical observations, we find that the optimal angles for multi-round GM-Th-QAOA are straightforward. Specifically, we present a simple scheme to determine the minimum number of rounds p necessary to achieve the maximum approximation ratio for a problem instance I and threshold th (this is akin to Grover's overshooting problem). For the first $p-1$ rounds, setting $\beta = \gamma = \pi$ gives optimal results, and we present an analytic formula for the optimal final angles β_p, γ_p .

We begin by describing the optimal angles for round p , followed by the method of determining the ideal number of rounds p . For the first $p-1$ rounds, we set $\beta = \gamma = \pi$ and denote the coefficients with the notation $c_{i,\pi}^{(j)}$ ($j \leq p-1, i \in \{0, 1\}$). The coefficients are then given by

$$\begin{aligned} c_{i,\pi}^{(j < p)} &= (-1)^i c_{i,\pi}^{(j-1)} - 2 \left(r c_{0,\pi}^{(j-1)} - (1-r) c_{1,\pi}^{(j-1)} \right), \\ c_0^{(p)} &= c_{0,\pi}^{(p-1)} - (1 - e^{-i\beta}) \left(r c_{0,\pi}^{(p-1)} + (1-r) c_{1,\pi}^{(p-1)} e^{-i\gamma} \right). \end{aligned} \quad (15)$$

Using the fact that $c_{i,\pi}^{(j < p)} \in \mathbb{R}$, we can solve $\|c_0^{(p)}\|^2 = 0$ to find the optimal angles

$$\begin{aligned} \beta_p &= \arctan \left(-\Delta \left| c_{0,\pi}^{(p-1)} \right|, \frac{2(1-r)}{|S|} - \left(c_{0,\pi}^{(p-1)} \right)^2 \right), \\ \gamma_p &= \arctan \left(-\frac{\Delta}{c_{1,\pi}^{(p-1)} \text{sgn}(c_{0,\pi}^{(p-1)})}, \frac{c_{0,\pi}^{(p-1)}(1-2r)}{c_{1,\pi}^{(p-1)}} \right), \end{aligned} \quad (16)$$

with

$$\Delta = \sqrt{4(1-r)|S|^{-1} - \left(c_{0,\pi}^{(p-1)}\right)^2}. \quad (17)$$

Our main insight is now that Eq. (16) only gives $\beta_p, \gamma_p \in \mathbb{R}$ when $\Delta \in \mathbb{R}$, i.e.

$$4(1-r) - |S| \left(c_{0,\pi}^{(p-1)}\right)^2 > 0. \quad (18)$$

If this inequality is not satisfied, then we have numerically observed that $\beta = \gamma = \pi$ again gives the best possible approximation ratio at round p , and further rounds are necessary to reach $\|c_0\|^2 = 0$.

The inequality in Eq. (18) does not actually have any dependence on $|S|$, as Eq. (15) shows that $c_{0,\pi}^{(j < p)} = |S|^{-1/2} \times (\text{a polynomial in } r)$. Therefore, for a given p , we can explicitly evaluate the recursion in Eq. (15) and determine the range of r values for which Eq. (18) is satisfied, thus determining the number of rounds necessary to get $\|c_0^{(p)}\|^2 = 0$ as a function of r :

- 1 round: $0 \leq r < 3/4$
- 2 rounds: $3/4 \leq r < \frac{1}{8}(\sqrt{5} + 5)$
- 3 rounds: $\frac{1}{8}(\sqrt{5} + 5) \leq r < 0.950484 \dots$
- 4 rounds: $0.950484 \dots \leq r < 0.969846 \dots$

We remark that the above is not a strict induction proof, as one could imagine a scenario where choosing a non- π set of angles at some round j allows for a greater approximation ratio (i.e. smaller $\|c_0^{(p)}\|^2$) to be reached in round p . However, in practice, this does not ever seem to be the case, as confirmed with a large number of numerical simulations.

C. Completing the Picture: Angle Finding

Recall that an input tuple for the quantum subroutine of GM-Th-QAOA is $(I, th, |\psi\rangle, H_P, H_{GM}, p, \vec{\beta}, \vec{\gamma})$. If we are given only a candidate threshold th , we find close-to-optimum angles $\vec{\beta}, \vec{\gamma}$ in $O(\log p)$ steps: The idea is to find a transition round t such that we have angles $\beta_i = \gamma_i = \pi$ for rounds $i < t$ and $\beta_i = \gamma_i = 0$ for rounds $t < i \leq p$:

- 1) Exponential search over the number of rounds $p_k = \lceil \lambda^k \rceil$ for small $\lambda > 1$, always using angles $\beta = \gamma = \pi$, until we find two values $k+1, k+2$ for which the p_{k+1} rounds give a higher expectation value than p_{k+2} rounds.
- 2) Do a modified binary search over the number of rounds p' in the interval $[p_k, p_{k+2}]$, to find maximum expectation value of a p' -round QAOA with angles $\beta = \gamma = \pi$.
- 3) The found p' could be an overshoot or an undershoot of the optimum number of rounds, and tightly bounds the range of possible values for r . Hence we test both a $t = p'$ -round and a $t = p' + 1$ -round QAOA with a linear search in r , with angles given by Eq. (16).

Our algorithm bears resemblance to the exponential quantum search algorithm [4] generalizing Grover's search to an unknown number of marked items. Depending on the number of rounds and desired accuracy, the final step may not be necessary, as setting $\beta = \gamma = \pi$ for all rounds gives near-optimal results. For small p it will also be more efficient to do a linear search over the number of rounds.

D. Completing the Picture: Threshold Finding

If we are given only $(I, |\psi\rangle, H_P, H_{GM}, p)$, we find the optimum threshold value th^* by doing a modified binary search on threshold values th with $0 \leq th \leq M$, where M is the maximum possible threshold value. For each value th that we test, we run the angle-finding algorithm from Section III-C.

This simple search technique is based on the results of extensive numerical experiments, which showed that for a given number of rounds p the approximation ratio as a function of th increases monotonically up to a peak value and then monotonically decreases. Should such a single-peak observation ever fail, we can always perform a linear search instead, increasing the number of steps from $O(\log M)$ to $O(M)$. Fig. 1 (left) shows the single peak for a random problem over 5 rounds.

We again observe a resemblance to a quantum search algorithm: The maximum finding algorithm [8] keeps increasing a threshold th which it passes to the mentioned exponential quantum search algorithm [4] to sample a new, higher threshold. This technique finds a maximum with a quadratic Grover speed-up over classical brute-force search. Our algorithm behaves somewhat similarly in case the number of rounds p is large enough to accommodate a meaningful use of large thresholds, see Fig. 1 (right). Furthermore, our approach allows for a potential increase in th round-over-round, which we leave to a future work.

In terms of actual implementation on all our algorithms, we can always get the full statevector as long as we just simulate classically; however, on actual quantum computers with measurements and returns of single basis states only, we need to execute enough rounds to arrive at a good approximation of the expectation value. Standard statistics using Chebyshev's inequality tells us that $O(M^3)$ shots are sufficient to approximate the expectation value within an additive error of 1 (see, e.g., [7] for a derivation in the QAOA context).

IV. EFFICIENT CLASSICAL NUMERICAL EVALUATION OF GROVER MIXER-BASED CIRCUITS

In preparation for our numerical experiments, we describe an efficient way to simulate the Grover Mixer QAOA for standard phase separators [5], [6] and the Threshold-based GM-Th-QAOA. The main source of our relative scalability relies on leveraging the Grover Mixer feature that solutions which evaluate to the same objective value also have identical amplitudes (thereby also discarding all infeasible basis states). Our approach calculates the expectation value only, as opposed to computing full statevector amplitudes, but this is sufficient.

We recall the definition (3) of the Grover Mixer [5], $e^{-i\beta|S\rangle\langle S|}$, which requires an efficient state preparation unitary preparing all feasible solutions S in equal superposition: $|S\rangle := |S|^{-1/2} \sum_{x \in S} |x\rangle$. For most of our experiments, this starting state is the Dicke state

$$|D_k^n\rangle := \binom{n}{k}^{-1/2} \sum_{HW(d_i)=k} d_i, \quad (20)$$

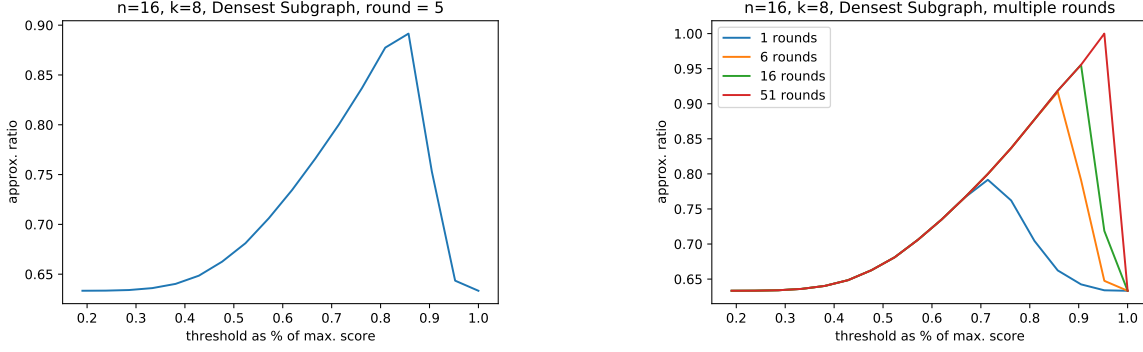


Fig. 1: Approximation ratio as a function of the threshold th (fraction of max score) for GM-Th-QAOA for k -Densest Subgraph: (left) shows the results for just a 5-round GM-Th-QAOA, (right) shows the evolution over many rounds for the same problem. This shows that at a given number of rounds p , the optimal threshold value increases monotonically up to a peak value and then decreases sharply. Furthermore, the threshold which produces the peak value for a given round p increases as p increases.

$$\begin{aligned}
 U_M(\beta_p) e^{-i\gamma_p H_P} |S^{(p-1)}\rangle &= (Id - (1 - e^{-i\beta_p}) |D_k^n\rangle \langle D_k^n|) \left(\sum_i c_i^{(p-1)} e^{-i\gamma_p h_i} |d_i\rangle \right) \\
 &= \sum_i \left[c_i^{(p-1)} e^{-i\gamma_p h_i} - \frac{1}{\binom{n}{k}} (1 - e^{-i\beta_p}) \left(\sum_j c_j^{(p-1)} e^{-i\gamma_p h_j} \right) \right] |d_i\rangle. \quad (19)
 \end{aligned}$$

Fig. 2: Explicit derivation of coefficients for fast numerical evaluation

which is the equal superposition of all computational basis states d_i on n qubits with exactly k one values, or more formally of Hamming weight k . Quantum circuits that prepare $|D_k^n\rangle$ deterministically in depth $O(n)$ exist [2]. After executing $p-1$ rounds of QAOA, we get the state

$$|S^{(p-1)}\rangle := \sum_i c_i^{(p-1)} |d_i\rangle, \quad (21)$$

where the sum is over all the individual states appearing in the Dicke state. For the p -th round, we begin by applying the phase separator:

$$e^{-i\gamma_p H_P} |S^{(p-1)}\rangle = \sum_i c_i^{(p-1)} e^{-i\gamma_p h_i} |d_i\rangle, \quad (22)$$

where $H_P |d_i\rangle = h_i |d_i\rangle$. We then then apply the Grover Mixer

$$U_M(\beta_p) = I - (1 - e^{-i\beta_p}) |D_k^n\rangle \langle D_k^n|, \quad (23)$$

to our state $e^{-i\gamma_p H_P} |S^{(p-1)}\rangle$, yielding Eq. 19 in Fig. 2. Beginning with $c_i^{(0)} = \binom{n}{k}^{-1/2}$, we can thus directly calculate the coefficients of the p -th round straightforwardly in terms of the coefficients of the $(p-1)$ -th round. However, it is unnecessary to calculate the coefficient for every computational basis state, as the Grover Mixer ensures that all states with the same objective function value have the same coefficient. In other words, if $h_i = h_j$, then $c_i = c_j$ for all rounds. Therefore, once we have precomputed the set of possible values for $H_P |d_i\rangle$ and their degeneracies (i.e., how many states share the same objective value), we can significantly reduce the total number of calculations necessary as follows: let us assume that the set of all $\{h_i\}$ can be reduced to the set of l distinct values

$\{g_i\}$ with degeneracies $\{d_i\}$. In many cases, the number of distinct objective function values is much smaller than the number of feasible solutions, $l \ll \binom{n}{k}$. For example, with k Densest Subgraph, l is $\mathcal{O}(n^2)$. If we are only interested in the Hamiltonian expectation value, $\langle S^{(p)} | H_P | S^{(p)} \rangle$, we need to only keep track of l coefficients, reducing our computational load considerably. If we label our reduced set of l coefficients $\{\hat{c}_i^{(p)}\}$, i.e. $\hat{c}_i^{(p)}$ is the coefficient for all states with objective cost g_i , we get from Eq. (19) to

$$\begin{aligned}
 \hat{c}_i^{(p)} &= \hat{c}_i^{(p-1)} e^{-i\gamma_p g_i} - \frac{(1 - e^{-i\beta_p})}{\binom{n}{k}} \sum_{j=1}^l d_j \hat{c}_j^{(p-1)} e^{-i\gamma_p g_j}, \\
 \Rightarrow \langle S^{(p)} | H_P | S^{(p)} \rangle &= \sum_{i=1}^l d_i g_i \left\| \hat{c}_i^{(p)} \right\|^2. \quad (24)
 \end{aligned}$$

V. EXPERIMENTAL EVALUATION

To test the performance of GM-Th-QAOA and compare it against standard GM-QAOA with regular phase separators, we execute a large set of experiments. Table I gives an overview of our experimental design. Our experiments were executed on a mid-range 64 core machine and took a few weeks; our software stack relies on standard numerical libraries using SciPy, NumPy in Python, and Julia. For each graph parameter combination (that is, the vertex count n and edge probabilities), we generate 30 random graphs with corresponding edge probabilities. We executed all our runs with the different numbers of rounds p by selecting values up to 16384. As phase separators, we used the threshold version, which is

Parameters	Optimization Problems and Grover Mixers	Phase Separators
$n = 15, 20, 35, 40, 50, 60, 70, 80, 90, 100$	k-Densest Subgraph — $ D_k^n\rangle \langle D_k^n $	Threshold
$k = .25n, .5n, .75n, n - 10$	k-Vertex Cover — $ D_k^n\rangle \langle D_k^n $	Standard
Edge Probabilities = .25, .5, .75	Max Cut — $ +^n\rangle \langle +^n $	
Number of rounds p : up to 16k	Max Bisection — $ D_{n/2}^n\rangle \langle D_{n/2}^n $	

TABLE I: Experimental Design: We performed numerical experiments with four different optimization problems and their corresponding Grover Mixers, comparing the standard and threshold phase separators. All input graphs were random graphs with edge probabilities from three different values. We studied graphs of up to 90 nodes, with k -values that define feasible solutions varied as three different fractions of node counts, as well as a standard $k = n - 10$ case. We generated 30 random graphs for each combination.

the key building block of GM-Th-QAOA and the standard Hamiltonian phase separator formulation.

For each run of GM-Th-QAOA, we calculate the optimal outer loop parameters th, β, γ as described in Section III through a set of test runs. For the standard phase separator GM-QAOA runs, we use the basin hopping technique to find good β, γ values, which is standard for QAOA simulations, while using similar precomputation and fast computation tricks as those described in Section IV. We set most basin hopping parameters to default values and observed that basin hopping converges to its optimum. For larger instances (i.e., $n > 20$), basin hopping becomes prohibitive in computational cost. Thus, we could not run all standard phase separator cases for high n and p counts, as evidenced in the plots. As other authors have observed [7], [19], our confidence that basin hopping finds near-optimum angle values is based among other factors on the fact that for a small number of rounds (< 6) with standard phase separators, basin hopping matches or beats the approximation ratios found through an exhaustive fine grid search. Thus, while it is possible that the truly best angles were not found by basin hopping for a few cases, our conclusions from the relative ranking of the two methods are sound.

We studied the following four graph optimization problems for input graphs $G = (V, E)$ with $n := |V|$ vertices and $m := |E|$ edges. They are all NP -hard, and have straightforward representations as Hamiltonians, see e.g., [14]:

- a) *Max k -Densest Subgraph*: Given an additional input parameter $k < n$, find a subset $V' \subset V$ of k vertices with a maximum number of edges between vertices from V' .
- b) *Max k -Vertex Cover*: Given an additional input parameter $k < n$, find a subset $V' \subset V$ of k vertices with a maximum number of edges with at least one end point in V' .
- c) *Max Cut*: Find a subset $V' \subset V$ of vertices with a maximum number of edges with one endpoint in V' and the other in $V \setminus V'$. This is an unconstrained problem, thus every computational basis state represents a feasible solution.
- d) *Max Bisection*: Find a subset $V' \subset V$ of $n/2$ vertices with a maximum number of edges with one end point in V' and the other in $V \setminus V'$.

Finally, we note that in the following comparisons we compare performance of GM-Th-QAOA and GM-QAOA given equivalent number of rounds p . This should not necessarily

be viewed as comparison given equivalent quantum resources, as we have not given an explicit construction of the threshold operator. We leave a more detailed discussion of the computational cost of these approaches, as well as other mixers, to a future work.

A. GM-Th-QAOA Outperforms Standard GM-QAOA

Our main result is that the Threshold-based GM-Th-QAOA finds better solutions as characterized by the approximation ratio across all optimization problems, graph sizes, edge probabilities, k parameter values, and number of rounds. We did not expect such a clean sweep.

Figures 3, 4, 5, 6, and 7 show a select set of example plots for different values of edge probabilities, k values and graph sizes for the four optimization problems. Each plot shows approximation ratio vs. the number of rounds for both the Threshold GM-Th-QAOA (red) and Standard GM-QAOA (blue), which means, of course, using the standard phase separator formulation as a Hamiltonian. The lighter shaded areas in red and blue show the minimum and maximum values found over the 30 random graphs tested. We use a different set of random graphs for each set of rounds in the plots, which explains the occasional surprising dip in performance at higher round counts. Both QAOA versions use the Grover mixer [5]. The plots in these figures show a consistent outperformance of Threshold over Standard QAOA, albeit the relative improvement varies between less than one percent (for k-Vertex Cover with edge probability of 0.75 and $k = 15$) to around 8 percent (for k-Vertex Cover with edge probability of 0.25 and $k = 5$). The advantage of Threshold QAOA remains constant in almost all cases as we increase the number of rounds. These same general statements hold across all parameter combinations from the experimental design defined in Table I; we omit such additional plots as they look very similar to the selected plots presented here.

We find this consistent outperformance to be somewhat astonishing. We emphasize that the outperformance is not due to the more efficient outer parameter search that GM-Th-QAOA allows for: all values on the plots are for the optimum outer parameter values. In fact, GM-Th-QAOA will not only find better solutions, but it will also find them faster because

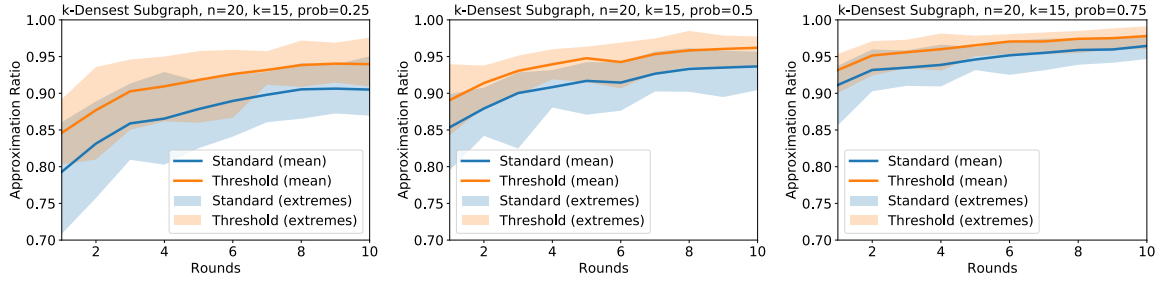


Fig. 3: **Maximum k -Densest Subgraph:** QAOA performance for different edge probabilities

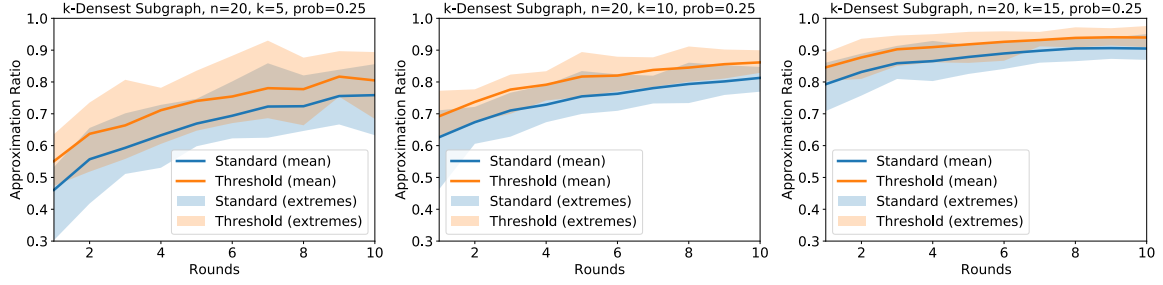


Fig. 4: **Maximum k -Densest Subgraph:** QAOA performance for different k -values

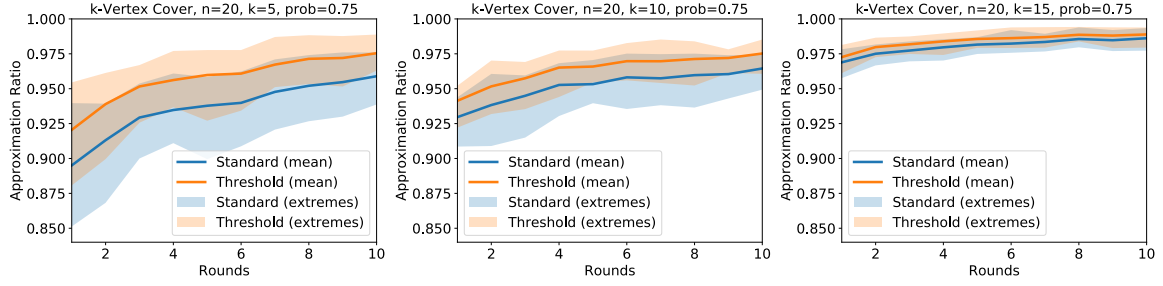


Fig. 5: **Maximum k -Vertex Cover:** QAOA performance for different k -values

the outer loop parameter value finding is also much simpler for GM-Th-QAOA than for Standard GM-QAOA.

While most of these plots tell a very similar story and mainly serve to drive home the point of how our main finding is valid across a large number of parameter values, we still find a few interesting differences: Figure 3 shows our two QAOA versions for Maximum k -Densest Subgraph with increasing edge probabilities (0.25, 0.50, 0.75 from left to right) on graphs of size 20 with k -value set to 15. Threshold and Standard QAOA both improve their performance as edge probability is increased, and the relative performance difference actually becomes a little smaller as we increase edge probability. We also note that the curves flatten out at round count of 8 or higher.

In contrast, Figure 4 shows the performance for $k = \{5, 10, 15\}$ again for graphs of size 20 and with edge probability fixed to 0.25. Increasing k again lead to improved performance and a narrower performance gap between the two QAOA versions.

Figure 5 shows performance results for Maximum k -Vertex Cover of graphs of size 20 with edge probability of 0.75 for different k -values. Threshold QAOA's performance advantage is less pronounced than it is for Maximum k -Densest Subgraph (note the y-axis values), but it still exists consistently.

Figure 6 studies the performance on the Maximum Bisection problem for graphs of size 15 (top row) and size 20 (bottom row) with increasing edge probabilities from left to right. The relative performance advantage of Threshold QAOA appears to be stable across the different graph sizes.

Figure 7 compares the QAOAs for Maximum Cut. Maximum Cut is the only unconstrained problem we considered, meaning all computational basis states are feasible solutions. Unlike other studies for MaxCut that limit node degrees to a small number, we use general random graphs. Thus the speed-up trick for constrained problems of only looking at feasible solutions no longer applies here. Simulation times thus start to get prohibitive a bit earlier here. Our plots show again that GM-Th-QAOA consistently outperforms GM-QAOA.

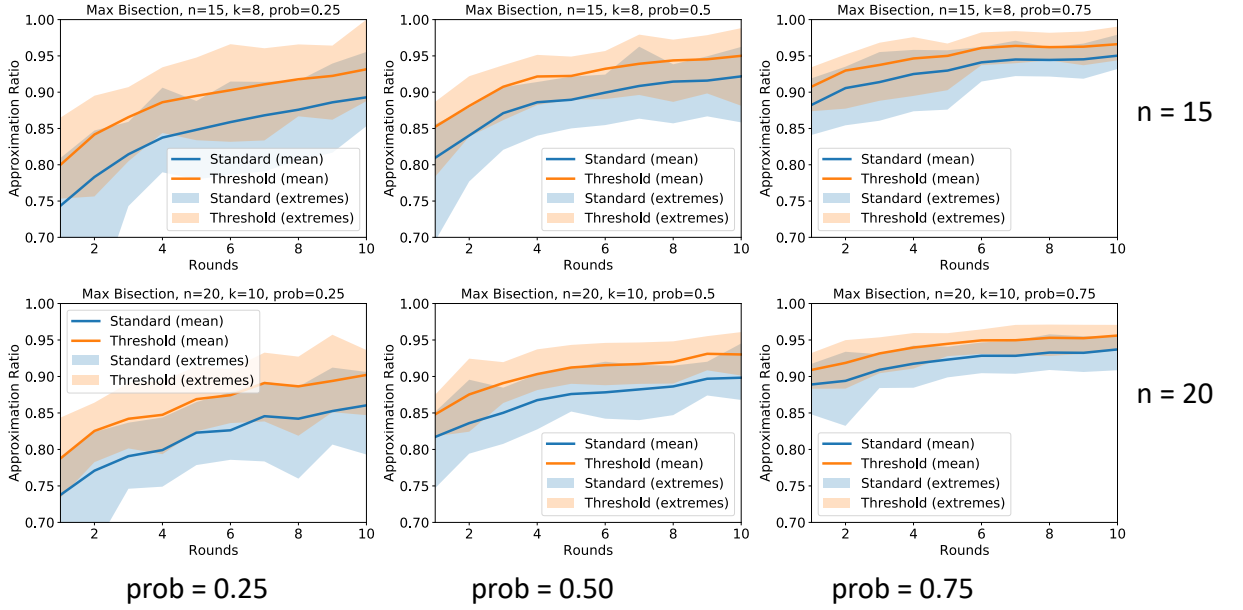


Fig. 6: **Maximum Bisection**: QAOA performance for graphs of size 15 (top row) and size 20 (bottom row) with varying edge probabilities (prob = 0.25, 0.50, 0.75 from left to right)

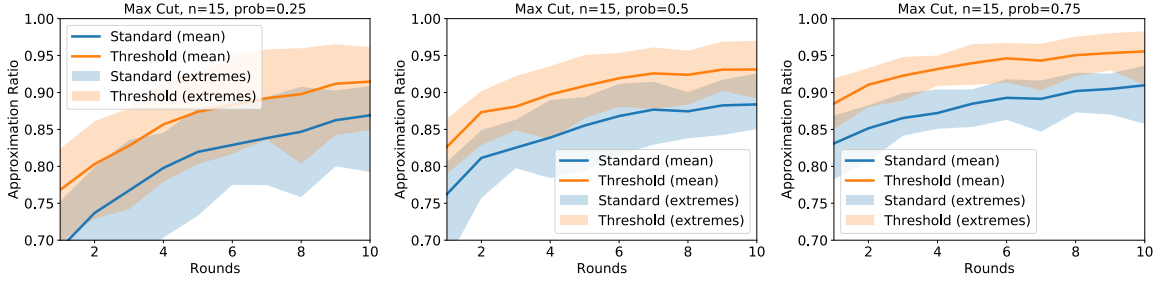


Fig. 7: **Maximum Cut**: QAOA performance for different edge probabilities

B. GM-Th-QAOA Simulation Speed-up Tricks Allow Scaling to 100 qubits and 16,384 rounds

Our tricks to speed up the numerical simulation of Threshold QAOA with the Grover Mixer enable us to classically simulate graph problems of 100 vertices, to near unlimited number of rounds. Such scaling greatly improves the confidence in our results, particularly with respect to ranking different QAOA variations and with respect to levels of approximation ratios achievable. Standard QAOA simulations so far (including our own) have mostly included simulations of graphs of up to 16 vertices and up to 10 rounds with performance and relative performance getting extrapolated from these small cases. A notable exception is a 25-round simulation of GM-QAOA for Maximum Satisfiability on 6 variables [1].

Figure 8 (left) shows the difference of the two QAOA versions for the k -Densest Subgraph problem on a medium-sized graph of 40 nodes with edge probability 0.5. Clearly, Threshold QAOA outperforms Standard QAOA, and here the performance gap appears to even increase with the number

of rounds. More importantly, however, classical simulation of Standard GM-QAOA becomes prohibitive on standard mid-range computing hardware, namely on the order of a few minutes per graph vs. milliseconds for GM-Th-QAOA.

Figure 8 (right) only shows GM-Th-QAOA runs for a single graph at each graph size up to $n = 100$ with edge probability 0.5 and $k = n - 10$. To show the effect of different number of rounds p allowed, we draw lines for various values of p . Clearly, a high round count results in better approximation ratios. The perhaps surprising effect that approximation ratios achieved improve with graph size n for smaller round counts can be explained through the edge density: a randomly selected subset of $n - 10$ vertices of a large graph will be closer to the optimum $n - 10$ subset than that of a smaller graph.

C. GM-Th-QAOA and the Grover speed-up

Threshold QAOA is a general-purpose quantum optimization heuristic with a subroutine runtime linear in the number of rounds p , which is usually chosen to be small. When combined with the Grover Mixer [1], [5], we have observed

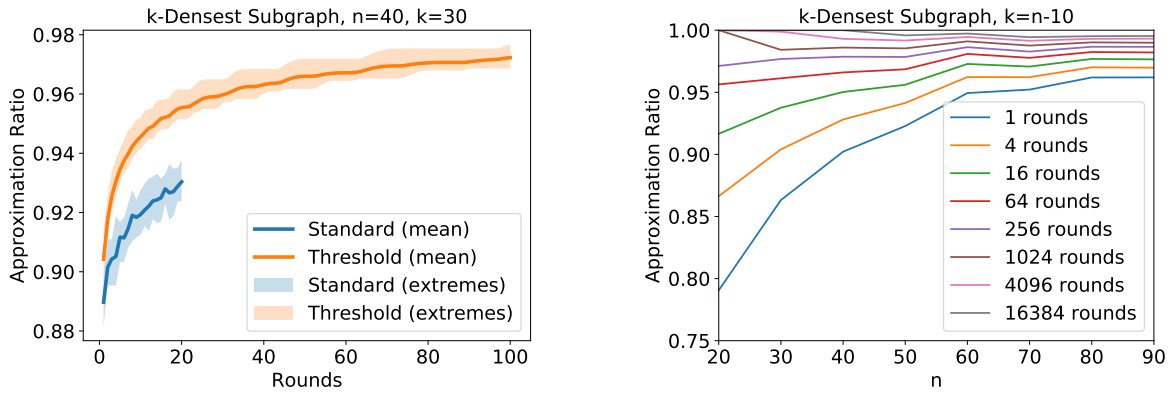


Fig. 8: Approximation ratios for the k -Densest Subgraph problem at different scales: (left) k -Densest Subgraph at intermediate scale of $n = 40$, $k = 30$. GM-Th-QAOA still outperforms standard GM-QAOA, but the parameter search of our simulation of GM-QAOA reaches its limits at 20 rounds. (right) Large-scale performance of GM-Th-QAOA for up to $n = 100$, $p = 2^{14}$.

that GM-Th-QAOA shares some traits with quantum search algorithms, such as sharing phase shift angles π (i.e., phase inversions) with the Grover oracle and diffusion operators [12], deploying an exponential quantum search [4] (Section III-C) or adapting a threshold value as in the minimum/maximum finding algorithm [8] (Section III-D).

In particular, for a very high number of rounds p (compared to the set of feasible solutions S , e.g. $p \in \Omega(\sqrt{|S|})$), the behaviour of our threshold and angle finding subroutines show a similar behaviour as the standard quantum minimum/maximum finding and exponential search algorithms. We do not endorse turning round count p into a variational parameter for GM-Th-QAOA. However, the thought experiment shows that GM-Th-QAOA will never fare worse than the minimum/maximum finding variant of Grover’s quantum search and thus also achieve at least a quadratic speed-up.

Thus, to QAOA advocates, this connection can be viewed as a formal proof that QAOA achieves at least some speed-up in an asymptotic setting. At the same time, QAOA skeptics will point to their long-standing assertion that QAOA will not outperform Grover search in a worst-case scenario. We have not found a formal proof to refute the skeptics’ claim, but rather we have designed a pragmatic QAOA heuristic that allows us to fix a (quantum) computational time budget and find the best possible solution within such a budget, which we believe will be of practical value once error-corrected and scalable quantum computing arrives.

VI. RELATED AND FUTURE WORK

Grover’s quantum search algorithm [12] was the first quantum algorithm to promise a quadratic speed-up over combinatorial *search* problems. When it comes to combinatorial *optimization* problems, we may be interested in either searching for the exact optimum, or approximating the problem with a close-to-optimal solution. The first approach exact optimization was a minimum finding algorithm [8] based on the quantum exponential search algorithm [4], which itself is a generalization of Grover search to an unknown number

of solutions. A different proposal for exact optimization was given in the form of the adiabatic algorithm [11].

Farhi et. al. later introduced the Quantum Approximate Optimization Algorithm [9] which can be seen as a low-order Trotterization of the adiabatic algorithm. Surprisingly, the minimum finding algorithm has not seen such an adaption to approximate optimization. To the best of our knowledge, this work is the first to close this gap, in the sense of Section V-C.

On the other hand, approaches using selective phase shift versions of Grover’s oracle and diffusion operators have been discussed for a while and exclusively in the search setting; starting with Grover’s fixed point quantum search with phase shifts of $\pm\pi/3$ [13], to more general angles which can recover the quadratic speed-up while preserving convergence to a fixed interval [20], to fully variational Grover search proposals which may give a moderate increase in success probability [17] while still (only) reflecting a Grover scaling [1]. A combination of a selective phase shift oracle with the transverse field based X -mixer in the search setting has been shown to achieve a quadratic speed-up as well [15], with success probability and number of rounds a constant factor away from the optimum query complexity of Grover search [21].

Finally, the selective-phase shift Grover diffusion operator has been introduced to the Quantum Alternating Operator Ansatz as the Grover Mixer for both unconstrained [1] and constrained [6] approximate optimization problems (albeit only in conjunction with standard phase separation), where it has been shown to moderately outperform the transverse field based X -mixer on unconstrained [1] and the XY -model Ring mixer on Hamming weight constrained problems [5], [7].

For future work, it will be important to give a full comparison of both Standard and Threshold phase separators in combination with all known problem-specific mixers. However, new insights in order to scale up problem sizes for simulations of parameter combinations beyond GM-Th-QAOA will be a necessary prerequisite for such future studies.

REFERENCES

- [1] V. Akshay, H. Philathong, M. E. S. Morales, and J. D. Biamonte. Reachability Deficits in Quantum Approximate Optimization. *Physical Review Letters*, 124(9):090504, 2020. [arXiv:1906.11259](#), doi: [10.1103/PhysRevLett.124.090504](#).
- [2] Andreas Bärtschi and Stephan Eidenbenz. Deterministic Preparation of Dicke States. In *22nd International Symposium on Fundamentals of Computation Theory, FCT'19*, pages 126–139, 2019. [arXiv:1904.07358](#), doi: [10.1007/978-3-030-25027-0_9](#).
- [3] Charles H. Bennett, Ethan Bernstein, Gilles Brassard, and Umesh Vazirani. Strengths and Weaknesses of Quantum Computing. *SIAM Journal on Computing*, 26(5):1510–1523, 1997. [arXiv:quant-ph/9701001](#), doi: [10.1137/S0097539796300933](#).
- [4] Michel Boyer, Gilles Brassard, Peter Høyer, and Alain Tapp. Tight bounds on quantum searching. *Fortschritte der Physik: Progress of Physics*, 46(4-5):493–505, 1998. [arXiv:quant-ph/9605034](#), doi: [10.1002/\(SICI\)1521-3978\(199806\)46:4/5<493::AID-PROP493>3.0.CO;2-P](#).
- [5] Andreas Bärtschi and Stephan Eidenbenz. Grover Mixers for QAOA: Shifting Complexity from Mixer Design to State Preparation. In *IEEE International Conference on Quantum Computing & Engineering QCE'20*, pages 72–82, 2020. [arXiv:2006.00354](#), doi: [10.1109/QCE49297.2020.00020](#).
- [6] Andreas Bärtschi and Stephan Eidenbenz. Grover Mixer QAOA: Feasible State Superpositions, Convergence to Optimum, and Limitations, 2021. in preparation.
- [7] Jeremy Cook, Stephan Eidenbenz, and Andreas Bärtschi. The Quantum Alternating Operator Ansatz on Maximum k-Vertex Cover. In *IEEE International Conference on Quantum Computing & Engineering QCE'20*, pages 83–92, 2020. [arXiv:1910.13483](#), doi: [10.1109/QCE49297.2020.00021](#).
- [8] Christoph Durr and Peter Hoyer. A quantum algorithm for finding the minimum. *arXiv e-prints*, 1996. [arXiv:quant-ph/9607014](#).
- [9] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. A Quantum Approximate Optimization Algorithm. *arXiv e-prints*, 2014. [arXiv:1411.4028](#).
- [10] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. A Quantum Approximate Optimization Algorithm Applied to a Bounded Occurrence Constraint Problem. *arXiv e-prints*, 2014. [arXiv:1412.6062](#).
- [11] Edward Farhi, Jeffrey Goldstone, Sam Gutmann, and Michael Sipser. Quantum Computation by Adiabatic Evolution. *arXiv e-prints*, 2000. [arXiv:quant-ph/0001106](#).
- [12] Lov K. Grover. A Fast Quantum Mechanical Algorithm for Database Search. In *28th Annual ACM Symposium on Theory of Computing, STOC'96*, pages 212–219, 1996. doi: [10.1145/237814.237866](#).
- [13] Lov K. Grover. Fixed-Point Quantum Search. *Physical Review Letters*, 95(15):150501, 2005. doi: [10.1103/PhysRevLett.95.150501](#).
- [14] Stuart Hadfield, Zhihui Wang, Bryan O’Gorman, Eleanor G Rieffel, Davide Venturelli, and Rupak Biswas. From the quantum approximate optimization algorithm to a quantum alternating operator ansatz. *Algorithms*, 12(2):34, 2019. [arXiv:1709.03489](#), doi: [10.3390/a12020034](#).
- [15] Zhang Jiang, Eleanor G. Rieffel, and Zhihui Wang. Near-optimal quantum circuit for Grover’s unstructured search using a transverse field. *Physical Review A*, 95:062317, Jun 2017. [arXiv:1702.02577](#), doi: [10.1103/PhysRevA.95.062317](#).
- [16] Sami Khairy, Ruslan Shaydulin, Lukasz Cincio, Yuri Alexeev, and Prasanna Balaprakash. Reinforcement-learning-based variational quantum circuits optimization for combinatorial problems. *arXiv e-prints*, 2019. [arXiv:1911.04574](#).
- [17] Mauro E. S. Morales, Timur Tlyachev, and Jacob Biamonte. Variational learning of Grover’s quantum search algorithm. *Physical Review A*, 98(6):062333, 2018. [arXiv:1805.09337](#), doi: [10.1103/PhysRevA.98.062333](#).
- [18] Zhihui Wang, Stuart Hadfield, Zhang Jiang, and Eleanor G. Rieffel. Quantum approximate optimization algorithm for MaxCut: A fermionic view. *Physical Review A*, 97:022304, 2018. [arXiv:1706.02998](#), doi: [10.1103/PhysRevA.97.022304](#).
- [19] Zhihui Wang, Nicholas C. Rubin, Jason M. Dominy, and Eleanor G. Rieffel. XY mixers: Analytical and numerical results for the quantum alternating operator ansatz. *Physical Review A*, 101(1):012320, 2020. [arXiv:1904.09314](#), doi: [10.1103/PhysRevA.101.012320](#).
- [20] Theodore J Yoder, Guang Hao Low, and Isaac L Chuang. Fixed-Point Quantum Search with an Optimal Number of Queries. *Physical Review Letters*, 113(21):210501, 2014. [arXiv:1409.3305](#), doi: [10.1103/PhysRevLett.113.210501](#).
- [21] Christof Zalka. Grover’s quantum searching algorithm is optimal. *Physical Review A*, 60:2746–2751, Oct 1999. [arXiv:quant-ph/9711070](#), doi: [10.1103/PhysRevA.60.2746](#).
- [22] Leo Zhou, Sheng-Tao Wang, Soonwon Choi, Hannes Pichler, and Mikhail D Lukin. Quantum Approximate Optimization Algorithm: Performance, Mechanism, and Implementation on Near-Term Devices. *Physical Review X*, 10(2):021067, 2020. [arXiv:1812.01041](#), doi: [10.1103/PhysRevX.10.021067](#).